

# PyGrabber

---

PyGrabber is a Python script that implements a simple polling mechanism to grab data from a source system. It can be used as a starting point for building a custom data extraction tool that fits your specific needs.

## Getting started

1. Clone the repository:

```
git clone https://github.com/meirm/pygrabber.git
cd pygrabber
```

2. Install the requirements:

```
pip install -r requirements.txt
```

3. Customize the script:

Open `grabber.py` in your favorite text editor and modify the variables in the **init** method to match your environment and requirements. For example, you might need to change the `clusterpath` variable to point to the location of your data cluster, or change the `poll_interval` variable to control how often the script should poll for data.

4. Run the script:

```
python grabber.py
```

The script will start polling for data and writing it to a file in the data directory.

## Usage

PyGrabber uses a simple polling mechanism to grab data from a source system. The main logic of the script is contained in the `run` method, which is called by the **init** method when an instance of the `PyGrabber` class is created.

By default, the script polls for data every second and writes it to a file in the data directory. You can customize the polling interval by changing the `poll_interval` variable in the **init** method.

PyGrabber also supports several Unix signals that can be used to control its behavior. Here are the supported signals:

SIGTERM: stops the script, renames the jobs to "interrupted", and exits

SIGUSR1: dumps the current status of the script

SIGUSR2: pauses or resumes the polling loop

To send a signal to a running instance of PyGrabber, use the kill command with the PID of the Python process that's running the script. For example, to send a SIGTERM signal to a PyGrabber instance with PID 1234, run:

```
kill -SIGTERM 1234
```

## License

PyGrabber is released under the MIT License. See LICENSE for details.